CHRISTIAN-ALBRECHTS-UNIVERSITÄT ZU KIEL
Institut für Informatik, Arbeitsgruppe Theorie der Parallelität
Prof. Dr. K. Jansen, K.-M. Klein, M. Rau

7. Juni 2016

Probeklausur zur Vorlesung »Algorithmen und Datenstrukturen«

SS 2016

Name: ________________________

Matrikel-Nr.: ________________________

Aufgabe 1 (Sortieren)

10 Punkte

Sortieren Sie die Sequenz $[12, 3, 6, 8, 2, 9, 1, 4]$ mittels Mergesort. Geben Sie geeignete Zwischenschritte an. Die rekursiven Aufrufe auf den Teilfeldern sollen dabei erkennbar sein.

Aufgabe 2 (Rekurrenzgleichung)

20 Punkte

Die Funktion $T : \mathbb{N} \to \mathbb{N}$ sei für alle $n \in \mathbb{N}$ durch

$$T(0) = 1,$$
$$T(n) = T(n-1) + 2n - 1 \text{ für } n \geq 1$$

definiert. Finden Sie eine geschlossene Darstellung für T und beweisen Sie Ihr Ergebnis mit vollständiger Induktion.

Aufgabe 3 (Plateau-Problem)

20 Punkte

Geben Sie einen Algorithmus mit Laufzeit $\mathcal{O}(n \log n)$ für das folgende Problem an: Gegeben sei ein nichtleeres Array A der Größe n mit beliebigen Zahlen. Gesucht ist ein Element, das am häufigsten in A vorkommt.

Geben Sie außerdem ein Algorithmus mit Laufzeit $O(n)$ an, wenn das Array nur Zahlen $\leq n$ enthält.

Beschreiben Sie kurz ihre Idee und geben Sie den Algorithmus in Pseudocode an. Sie dürfen Algorithmen verwenden, die aus der Vorlesung bekannt sind.

Aufgabe 4 (AVL-Bäume)

30 Punkte

Gegeben sei ein AVL-Baum mit n Knoten. Beweisen Sie die folgende Schranke aus der Vorlesung: Die Höhe eines AVL-Baums ist durch $\approx 1.44 n \log n$ beschränkt.

Aufgabe 5 (Median bei Quicksort)

30 Punkte

Der Median einer unsortierten Folge $a_1, \ldots, a_n$ von n unterschiedlichen Zahlen ist diejenige Zahl, die an Stelle $\lfloor n/2 \rfloor + 1$ in der entsprechend sortierten Folge steht. Interessanterweise ist das Bestimmen des Medians einfacher als das Sortieren, d.h. es existiert ein Algorithmus, der den Median einer unsortierten Folge in Zeit $\mathcal{O}(n)$ bestimmt (ohne die Folge zu sortieren).

Gegeben sei nun ein Algorithmus, der den Median von einer Folge von n unterschiedlichen Zahlen mit $c \cdot n$ Operationen berechnet für eine Konstante $c \in \mathbb{N}$.

Wir betrachten nun den deterministischen Quicksort, der kein zusätzliches Feld in jeder Iteration benötigt und der nur Eingaben von Folgen mit unterschiedlichen Zahlen bekommt.
Wie ändert sich die Worst-Case Laufzeit $T(n)$ von Quicksort bei einer Eingabe von n unterschiedlichen Zahlen, wenn der Median immer als Pivotelement bestimmt und gewählt wird? Gehen Sie dazu in folgenden Schritten vor:

- Geben Sie eine Vermutung an, ob die Laufzeit im Worst-Case Fall weiterhin in $\mathcal{O}(n^2)$ liegt, oder besser wird.
- Bestimmen Sie die Rekurrenzgleichung für $T(n)$. Hierbei dürfen Sie annehmen, dass ein einelementiges Feld in c_0 Operationen abgearbeitet wird und ein Feld mit n Elementen neben der Bestimmung des Medians $c_1 \cdot n + c_2$ zusätzliche Operationen benötigt, für Konstanten $c_0, c_1, c_2 \in \mathbb{N}$.
- Finden Sie eine geschlossene Form für die Rekurrenzgleichung $T(n)$, für alle $n \in \mathbb{N}$ für die ein $k \in \mathbb{N}$ existiert und $n = 2^k - 1$ gilt.
- Finden Sie eine geeignete Abschätzung von $T(n)$ und ordnen Sie diese entsprechend der $\mathcal{O}$-Notation ein.

Viel Erfolg!

Die korrigierten Klausuren werden in den einzelnen Präsenzübungen zurückgegeben.